H 5

```
   SSSSSSSSSSSS     OOOOOOOOO     RRRRRRRRRRRR     TTTTTTTTTTTTTTT     333333333        222222222
   SSSSSSSSSSSS     OOOOOOOOO     RRRRRRRRRRRR     TTTTTTTTTTTTTTT     333333333        222222222
   SSSSSSSSSSSS     OOOOOOOOO     RRRRRRRRRRRR     TTTTTTTTTTTTTTT     333333333        222222222
SSS              OOO         OOO  RRR         RRR        TTT        333         333  222         222
SSS              OOO         OOO  RRR         RRR        TTT        333         333  222         222
SSS              OOO         OOO  RRR         RRR        TTT        333         333  222         222
SSS              OOO         OOO  RRR         RRR        TTT                    333              222
SSS              OOO         OOO  RRR         RRR        TTT                    333              222
SSS              OOO         OOO  RRR         RRR        TTT                    333              222
   SSSSSSSSS     OOO         OOO  RRRRRRRRRRRR           TTT                 333              222
   SSSSSSSSS     OOO         OOO  RRRRRRRRRRRR           TTT                 333              222
   SSSSSSSSS     OOO         OOO  RRRRRRRRRRRR           TTT                 333              222
            SSS  OOO         OOO  RRR   RRR              TTT                    333        222
            SSS  OOO         OOO  RRR   RRR              TTT                    333        222
            SSS  OOO         OOO  RRR   RRR              TTT                    333        222
            SSS  OOO         OOO  RRR      RRR           TTT        333         333     222
            SSS  OOO         OOO  RRR      RRR           TTT        333         333     222
            SSS  OOO         OOO  RRR      RRR           TTT        333         333     222
SSSSSSSSSSSS        OOOOOOOOO     RRR         RRR        TTT           333333333     222222222222222
SSSSSSSSSSSS        OOOOOOOOO     RRR         RRR        TTT           333333333     222222222222222
SSSSSSSSSSSS        OOOOOOOOO     RRR         RRR        TTT           333333333     222222222222222
```

```
_$2

Pse
---
SOR

SOR

SOR

SOR

_LI
```

```
**FILE**ID**SORCOLLAT                                    J 9



  SSSSSSSS    OOOOOO    RRRRRRRR      CCCCCCCC    OOOOOO    LL          LL            AAAAAA    TTTTTTTTTT
  SSSSSSSS    OOOOOO    RRRRRRRR      CCCCCCCC    OOOOOO    LL          LL            AAAAAA    TTTTTTTTTT
SS          OO      OO  RR      RR  CC          OO      OO  LL          LL          AA      AA      TT
SS          OO      OO  RR      RR  CC          OO      OO  LL          LL          AA      AA      TT
SS          OO      OO  RR      RR  CC          OO      OO  LL          LL          AA      AA      TT
SS          OO      OO  RR      RR  CC          OO      OO  LL          LL          AA      AA      TT
  SSSSSS    OO      OO  RRRRRRRR    CC          OO      OO  LL          LL          AA      AA      TT
  SSSSSS    OO      OO  RRRRRRRR    CC          OO      OO  LL          LL          AA      AA      TT
        SS  OO      OO  RR  RR      CC          OO      OO  LL          LL          AAAAAAAAAA      TT
        SS  OO      OO  RR  RR      CC          OO      OO  LL          LL          AAAAAAAAAA      TT
        SS  OO      OO  RR    RR    CC          OO      OO  LL          LL          AA      AA      TT           ....
        SS  OO      OO  RR    RR    CC          OO      OO  LL          LL          AA      AA      TT           ....
SSSSSSSS      OOOOOO    RR      RR    CCCCCCCC    OOOOOO    LLLLLLLLLL  LLLLLLLLLL  AA      AA      TT           ....
SSSSSSSS      OOOOOO    RR      RR    CCCCCCCC    OOOOOO    LLLLLLLLLL  LLLLLLLLLL  AA      AA      TT           ....

LL            IIIIII      SSSSSSSS
LL            IIIIII      SSSSSSSS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II        SSSSSS
LL              II        SSSSSS
LL              II              SS
LL              II              SS
LL              II              SS
LL              II              SS
LLLLLLLLLL    IIIIII    SSSSSSSS
LLLLLLLLLL    IIIIII    SSSSSSSS
```

Table of contents

```
    (1)        3        Copyright Notice
    (1)       29        Program description
```

```
                                                            L 9
SOR$COLLATE                Compare under collating sequence        16-SEP-1984 01:20:07  VAX/VMS Macro V04-00          Page   1
V04-000                                                             5-SEP-1984 03:36:01  [SORT32.SRC]SORCOLLAT.MAR;1          (1)

                         0000          .TITLE  SOR$COLLATE                   Compare under collating sequence
                         0000          .IDENT  'V04-000'                     ; File: SORCOLLAT.MAR Edit: PDG022
                         0000          .SBTTL  Copyright Notice
                         0000  ;
                         0000  ;****************************************************************************
                         0000  ;*                                                                          *
                         0000  ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
                         0000  ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
                         0000  ;*  ALL RIGHTS RESERVED.                                                    *
                         0000  ;*                                                                          *
                         0000  ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
                         0000  ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
                         0000  ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
                         0000  ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
                         0000  ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
                         0000  ;*  TRANSFERRED.                                                            *
                         0000  ;*                                                                          *
                         0000  ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
                         0000  ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
                         0000  ;*  CORPORATION.                                                            *
                         0000  ;*                                                                          *
                         0000  ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
                         0000  ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
                         0000  ;*                                                                          *
                         0000  ;*                                                                          *
                         0000  ;****************************************************************************
                         0000  ;
```

```
SOR
Sym

EQ
GT
JOI
JOI
LT
NE
POP
PSW
PSW
PSW
RES
RES
RES
RES
RES
RES
SOR
SOR
SOR
SOR
SOR
USE
USE


PSE
---

.
$AB
SOR


Pha
---
Int
Com
Pas
Sym
Pas
Sym
Pse
Cro
Ass

The
114
The
470
17
```

```
0000            .SBTTL  Program description
0000    ;++
0000    ;
0000    ; FACILITY:
0000    ;
0000    ;       Support for compare under influence of collating sequences.
0000    ;
0000    ; ABSTRACT:
0000    ;
0000    ;       NONE
0000    ;
0000    ; ENVIRONMENT:
0000    ;
0000    ;       Native mode, User mode, AST reentrant
0000    ;
0000    ; AUTHOR:
0000    ;
0000    ;       Peter D Gilbert, January 1983
0000    ;
0000    ; MODIFIED BY:
0000    ;
0000    ;       T03-015         Original
0000    ;       T03-016 Support ignored pad characters.  PDG 28-Jan-1983
0000    ;       T03-017 Fix errors with PUSHR and POPR order.  PDG 1-Feb-1983
0000    ;       T03-018 Add XORB comments.  PDG 7-Mar-1983
0000    ;       T03-019 Reverse sense of the tie-breaking CMPC.  PDG 22-Mar-1983
0000    ;       T03-020 Make tie-break CMPC reversable at run-time.  PDG 12-Apr-1983
0000    ;       T03-021 Note that R5 is preserved.  PDG 11-Jul-1983
0000    ;       T03-022 Remove extra store of stable point.  Add "short-cut" entry
0000    ;               points.  PDG 17-Oct-1983
0000    ;--
```

```
SOR
VAX



Mac
---
_$2

76

The

MAC
```

```
    0000        .DSABL  GLOBAL          ; Externals must be explicitly declared
    0000        $PSWDEF
    0000
00000000        .PSECT  SOR$RO_CODE     NOVEC,NOWRT,RD,EXE,SHR,LCL,REL,CON,PIC,LONG
    0000  ;+
    0000  ;
    0000  ; These routinse do comparisons under the influence of a collating sequence.
    0000  ;
    0000  ; Inputs:
    0000  ;    R0  Length remaining in string1
    0000  ;    R1  Address of source1
    0000  ;    R2  Length remaining in string2
    0000  ;    R3  Address of source2
    0000  ;    R5  Address of tables
    0000  ;    0(SP)       Return address
    0000  ;
    0000  ; Outputs:
    0000  ;    R0  -1, 0, +1
    0000  ;    R1 thru R4  Garbage
    0000  ;    R5  Preserved
    0000  ;    R9  Preserved
    0000  ;    Condition codes as a result of MOVL R0, R0
    0000  ;    The return address is removed from the stack.
    0000  ;
    0000  ;--
```

```
0000             ; Fetch a character.
0000             ;
0000             .MACRO   FETCHSTR1, ?LAB1, ?LAB2
0000             SOBGEQ   R0, LAB1                      ; Any characters left?
0000             CLRL     R0                            ; Make this zero again
0000             MOVZBL   B^RES$PAD(R5), R4             ; Use the pad character
0000             BRB      LAB2
0000    LAB1:    MOVZBL   (R1)+, R4                     ; Fetch the character
0000    LAB2:    .ENDM
0000             .MACRO   FETCHSTR2, ?LAB1, ?LAB2
0000             SOBGEQ   R2, LAB1                      ; Any characters left?
0000             CLRL     R2                            ; Make this zero again
0000             MOVZBL   B^RES$PAD(R5), R9             ; Use the pad character
0000             BRB      LAB2
0000    LAB1:    MOVZBL   (R3)+, R9                     ; Fetch the character
0000    LAB2:    .ENDM
0000
0000             ; Fetch a character and look up its value in the primary table.
0000             ; If we were padding, and the pad character is ignored, leave the
0000             ; collating value as zero, so that this string will compare less than
0000             ; any non-ignored characters in the other string.
0000             ; If both string are being padded with ignored pad characters, the code
0000             ; branches back to the CMPC instruction, which compares equal, and the
0000             ; tie-break information will then be used.
0000             ;
0000             ; When we have the collating value, either fall through, or branch to
0000             ; the first parameter.
0000             ; The second parameter labels the call to SPECSTR or SPECDBL, so that
0000             ; other code can branch to it.
0000             ;
0000             .MACRO   FETCHSTR1NP, RESLAB, ?LAB1
0000             SOBGEQ   R0, LAB1                      ; Any characters left?
0000             CLRL     R0                            ; Make this zero again
0000             MOVZBL   B^RES$PAD(R5), R4             ; Use the pad character
0000             MOVZBL   B^RES$PTAB(R5)[R4], R4        ; Fetch the collating value
0000             BRB      RESLAB                        ; Use this value, regardless
0000    LAB1:    MOVZBL   (R1)+, R4                     ; Fetch the character
0000             MOVZBL   B^RES$PTAB(R5)[R4], R4        ; Check that it's not special
0000             BNEQ     RESLAB
0000             .ENDM
0000             .MACRO   FETCHSTR2NP, RESLAB, ?LAB1
0000             SOBGEQ   R2, LAB1                      ; Any characters left?
0000             CLRL     R2                            ; Make this zero again
0000             MOVZBL   B^RES$PAD(R5), R9             ; Use the pad character
0000             MOVZBL   B^RES$PTAB(R5)[R9], R9        ; Fetch the collating value
0000             BRB      RESLAB                        ; Use this value, regardless
0000    LAB1:    MOVZBL   (R3)+, R9                     ; Fetch the character
0000             MOVZBL   B^RES$PTAB(R5)[R9], R9        ; Check that it's not special
0000             BNEQ     RESLAB
0000             .ENDM
0000
0000             .MACRO   SPECSTR1, ?LAB1, ?LAB2, ?LAB3
0000             MOVAB    W^RES$STAB(R5), R4
0000             BRB      LAB2
0000    LAB1:    ADDL2    #4, R4
0000    LAB2:    CMPB     (R4), -1(R1)
0000             BLSSU    LAB1
```

```
0000                   BEQL    LAB3
0000                   MOVAL   #0, R4
0000          LAB3:    MOVZWL  2(R4), R4
0000                   .ENDM
0000                   .MACRO  SPECSTR2, ?LAB1, ?LAB2, ?LAB3
0000                   MOVAB   W^RES$STAB(R5), R9
0000                   BRB     LAB2
0000          LAB1:    ADDL2   #4, R9
0000          LAB2:    CMPB    (R9), -1(R3)
0000                   BLSSU   LAB1
0000                   BEQL    LAB3
0000                   MOVAL   #0, R9
0000          LAB3:    MOVZWL  2(R9), R9
0000                   .ENDM
0000
0000                   .MACRO  SPECDBL1, ?LAB1, ?LAB2, ?LAB3, ?LAB4
0000                   ;
0000                   ; This code assumes that the length of the original string is non-zero.
0000                   ;
0000                   MOVAB   W^RES$STAB(R5), R4          ; Get the table address
0000                   CLRL    -(SP)                       ; Assume it is ignored
0000                   BRB     LAB2
0000          LAB1:    ADDL2   #4, R4                      ; Advance past the collating value
0000          LAB2:    CMPB    (R4), -1(R1)                ; Does this first character match?
0000                   BLSSU   LAB1
0000                   BGTRU   LAB4
0000                   ADDL2   #2, R4
0000                   MOVW    (R4), (SP)                  ; It matched.  Save the collating value
0000                   TSTW    R0
0000                   BEQL    LAB4                        ; Any more chars in the string?
0000          LAB3:    ADDL2   #2, R4
0000                   CMPW    (R4)+, -1(R1)               ; Do the two characters match?
0000                   BLSSU   LAB3
0000                   BGTRU   LAB4
0000                   TSTW    2(R4)
0000                   BEQL    LAB4                        ; Branch if we found the trailing stuff
0000                   MOVW    (R4), (SP)                  ; Copy the collating value
0000                   DECL    R0                          ; Advance the string
0000                   INCL    R1
0000          LAB4:    MOVL    (SP)+, R4                   ; Put the collating value in R4
0000                   .ENDM
0000                   .MACRO  SPECDBL2, ?LAB1, ?LAB2, ?LAB3, ?LAB4
0000                   MOVAB   W^RES$STAB(R5), R9
0000                   CLRL    -(SP)
0000                   BRB     LAB2
0000          LAB1:    ADDL2   #4, R9
0000          LAB2:    CMPB    (R9), -1(R3)
0000                   BLSSU   LAB1
0000                   BGTRU   LAB4
0000                   ADDL2   #2, R9
0000                   MOVW    (R9), (SP)
0000                   TSTW    R2
0000                   BEQL    LAB4
0000          LAB3:    ADDL2   #2, R9
0000                   CMPW    (R9)+, -1(R3)
0000                   BLSSU   LAB3
0000                   BGTRU   LAB4
```

```
0000          TSTW    2(R9)
0000          BEQL    LAB4
0000          MOVW    (R9), (SP)
0000          DECL    R2
0000          INCL    R3
0000  LAB4:   MOVL    (SP)+, R9
0000          .ENDM
0000
0000          ; Define offsets
0000          ;
0000          .EXTRN  RES$PTAB                   ; Address of the primary table
0000          .EXTRN  RES$UPPER                  ; Address of the upper table
0000          .EXTRN  RES$TB                     ; Tie-break PSW
0000          .EXTRN  RES$REVERSE                ; Reverse sense of tie-break CMPC
0000          .EXTRN  RES$PAD                    ; Pad character
0000          .EXTRN  RES$STAB                   ; Address of secondary table
```

```
                                 0000  ;
                                 0000  ; SOR$$COLLATE_0
                                 0000  ;
                                 0000  ;   This routine assumes/supports:
                                 0000  ;       No ignored characters
                                 0000  ;       No double characters
                                 0000  ;       No double collating values
                                 0000  ;       Upcase table
                                 0000  ;
                                 0000  ; Registers:
                                 0000  ;       R0-R4   Nopreserve
                                 0000  ;       R5      Preserve
                                 0000  ;       R6-R8   Not used
                                 0000  ;       R9-R10  Preserve
                                 0000  ;       R11     Preserve
                                 0000  ;
                                 0000  SOR$$COLLATE_0_A::
                   0200 8F   BB  0000          PUSHR   #^M<R9>
                          7E DC  0004          MOVPSL  -(SP)
                          10 12  0006          BNEQ    JOIN_0
                          58 11  0008          BRB     USETB
                                 000A  SOR$$COLLATE_0::
                     7E   59 D0  000A          MOVL    R9, -(SP)                       ; Save R9
     63   52   00'A5  61  50 2D  000D          CMPC5   R0, (R1), B^RES$PAD(R5), R2, (R3)
                          7E DC  0014          MOVPSL  -(SP)                           ; Save the tie-break info
                          4A 13  0016          BEQL    USETB                           ; An optimization
               6E    00'A5   8C  0018  JOIN_0: XORB2   B^RES$REVERSE(R5), (SP) ; Reverse sense of this compare
               6E    00'A5   88  001C          BISB2   B^RES$TB(R5), (SP)      ; Tie-break adjustment
                                 0020  0$:     ;
                                 0020          ; Fetch the characters from the strings.
                                 0020          ;
                                 0020          FETCHSTR1
                                 002E          FETCHSTR2
                                 003C          ;
                                 003C          ; We know they differ.
                                 003C          ; Are they different cases of the same char?
                                 003C          ;
                  10 6E   01 E0  003C          BBS     #PSW$V_V, (SP), 16$
      0000'C549  0000'C544   91  0040          CMPB    W^RES$UPPER(R5)[R4], W^RES$UPPER(R5)[R9]
                          05 13  0049          BEQL    16$                             ; Yes, the binary compare is okay
                          6E DC  004B          MOVPSL  (SP)                            ; No, use this compare
                     6E   02 C8  004D          BISL2   #PSW$M_V, (SP)                  ; Don't do this again
                                 0050  16$:    ;
                                 0050          ; Compare the collating values of the characters.
                                 0050          ;
        00'A549    00'A544   91  0050          CMPB    B^RES$PTAB(R5)[R4], B^RES$PTAB(R5)[R9]
                          18 12  0057          BNEQ    POPNE
     63   52   00'A5  61  50 2D  0059          CMPC5   R0, (R1), B^RES$PAD(R5), R2, (R3)
                          BE 12  0060          BNEQ    0$                              ; If equal, then R0 = 0
                   0201 8F   BA  0062  USETB:  POPR    #^M<R0,R9>                      ; Pop PSL, Restore R9
                  11 50   02 E0  0066  USETB1: BBS     #PSW$V_Z, R0, EQ                ; Zero?
                     0A   50 E9  006A          BLBC    R0, GT                          ; Like BGTRU, since PSW$V_C = 0
                     50   01 CE  006D  LT:     MNEGL   #1, R0                          ; Return -1
                             05  0070          RSB
                   0201 8F   BA  0071  POPNE:  POPR    #^M<R0,R9>                      ; Pop PSL, restore R9
                          F6 1B  0075  NE:     BLEQU   LT
                     50   01 D0  0077  GT:     MOVL    #1, R0                          ; Return +1
```

```
                          05  007A    275             RSB
                  50      D4  007B    276 EQ:         CLRL    R0
                          05  007D    277             RSB
```

```
                             007E     ;
                             007E     ; SOR$$COLLATE_1
                             007E     ;
                             007E     ;   This routine assumes/supports:
                             007E     ;       Ignored characters
                             007E     ;       No double characters
                             007E     ;       Double collating values
                             007E     ;       Upcase table
                             007E     ;       Ignored pad characters
                             007E     ;
                             007E     ; Registers:
                             007E     ;       R0-R4   Nopreserve
                             007E     ;       R5      Preserve
                             007E     ;       R6-R8   Not used
                             007E     ;       R9-R10  Preserve
                             007E     ;       R11     Preserve
                             007E     ;
                             007E             .ENABL  LSB
                             007E     SOR$$COLLATE_1_A::
                  0200 8F    BB  007E         PUSHR   #^M<R9>
                       7E    DC  0082         MOVPSL  -(SP)
                       10    12  0084         BNEQ    JOIN_1
                       DA    11  0086         BRB     USETB
                             0088     SOR$$COLLATE_1::
                  7E   59    D0  0088         MOVL    R9, -(SP)                       ; Save R9
 63   52   00'A5  61   50    2D  008B         CMPC5   R0, (R1), B^RES$PAD(R5), R2, (R3)
                       7E    DC  0092         MOVPSL  -(SP)                           ; Save the tie-break info
                       CC    13  0094         BEQL    USETB                           ; An optimization
            6E   00'A5       8C  0096     JOIN_1: XORB2 B^RES$REVERSE(R5), (SP)      ; Reverse sense of this compare
            6E   00'A5       88  009A         BISB2   B^RES$TB(R5), (SP)              ; Tie-break adjustment
                             009E     0$:     ;
                             009E             ; Fetch the characters from strings
                             009E             ;
                             009E             FETCHSTR1
                             00AC             FETCHSTR2
                             00BA             ;
                             00BA             ; We know they differ.
                             00BA             ; Are they different cases of the same char?
                             00BA             ;
               10 6E   01    E0  00BA         BBS     #PSW$V_V, (SP), 16$
     0000'C549  0000'C544    91  00BE         CMPB    W^RES$UPPER(R5)[R4], W^RES$UPPER(R5)[R9]
                       05    13  00C7         BEQL    16$                             ; Yes, the binary compare is okay
                       6E    DC  00C9         MOVPSL  (SP)                            ; No, use this compare
                  6E   02    C8  00CB         BISL2   #PSW$M_V, (SP)                  ; Don't do this again
                             00CE     16$:    ;
                             00CE             ; Fetch the collating values of the characters.
                             00CE             ;
            54   00'A544     9A  00CE         MOVZBL  B^RES$PTAB(R5)[R4], R4
                       18    13  00D3         BEQL    101$
            59   00'A549     9A  00D5         MOVZBL  B^RES$PTAB(R5)[R9], R9
                       5A    13  00DA         BEQL    202$
                  59   54    91  00DC         CMPB    R4, R9
                       90    12  00DF     150$:   BNEQ    POPNE
 63   52   00'A5  61   50    2D  00E1     100$:   CMPC5   R0, (R1), B^RES$PAD(R5), R2, (R3)
                       B4    12  00E8         BNEQ    0$                              ; If equal, then R0 = 0
                     FF75    31  00EA         BRW     USETB
            59   00'A549     9A  00ED     101$:   MOVZBL  B^RES$PTAB(R5)[R9], R9
```

```
                  21   11  00F2   336          BRB     112$
 59    59     F8  8F   78  00F4   337 402$:    ASHL    #-8, R9, R9
                  E6   13  00F9   338          BEQL    100$
                           00FB   339 111$:    FETCHSTR1NP     113$
                           0115   340 112$:    SPECSTR1
                  C7   13  0132   341          BEQL    111$
                  59   95  0134   342 113$:    TSTB    R9
                  2D   13  0136   343 202$:    BEQL    201$
             59   54   91  0138   344 301$:    CMPB    R4, R9
                  A2   12  013B   345          BNEQ    150$
 54    54     F8  8F   78  013D   346          ASHL    #-8, R4, R4
                  B0   13  0142   347          BEQL    402$
 59    59     F8  8F   78  0144   348          ASHL    #-8, R9, R9
                  ED   12  0149   349          BNEQ    301$
                           014B   350 211$:    FETCHSTR2NP     301$
                           0165   351 201$:    SPECSTR2
                  C7   13  0182   352          BEQL    211$
                  B2   11  0184   353          BRB     301$
                           0186   354          .DSABL  LSB
```

```
                                   0186   356 ;
                                   0186   357 ; SOR$$COLLATE_2
                                   0186   358 ;
                                   0186   359 ;   This routine assumes/supports:
                                   0186   360 ;         Ignored characters
                                   0186   361 ;         Double characters
                                   0186   362 ;         Double collating values
                                   0186   363 ;         Upcase table
                                   0186   364 ;         Ignored pad characters
                                   0186   365 ;
                                   0186   366 ; Registers:
                                   0186   367 ;         R0-R4   Nopreserve
                                   0186   368 ;         R5      Preserve
                                   0186   369 ;         R6-R8   Not used
                                   0186   370 ;         R9-R10  Preserve
                                   0186   371 ;         R11     Preserve
                                   0186   372 ;
                                   0186   373 ; Restrictions:
                                   0186   374 ;         The following are not handled correctly:
                                   0186   375 ;             Double characters that use the pad character.
                                   0186   376 ;                 (when this happens at the end of a string)
                                   0186   377 ;
                                   0186   378         .ENABL  LSB
                                   0186   379 SOR$$COLLATE_2::
               0202 8F        BB   0186   380         PUSHR   #^M<R1,R9>                      ; Save R9, save current stable point
63   52   00'A5   61   50     2D   018A   381         CMPC5   R0, (R1), B^RES$PAD(R5), R2, (R3)
                       7E     DC   0191   382         MOVPSL  -(SP)                           ; Save the tie-break info
                       7A     13   0193   383         BEQL    89$                             ; An optimization
             6E   00'A5       8C   0195   384         XORB2   B^RES$REVERSE(R5), (SP)         ; Reverse sense of this compare
             6E   00'A5       88   0199   385         BISB2   B^RES$TB(R5), (SP)              ; Tie-break adjustment
                                   019D   386 0$:     ;
                                   019D   387         ; Backup to some stable point
                                   019D   388         ;
                  59   51     D0   019D   389         MOVL    R1, R9
             04   AE   51     D1   01A0   390 2$:     CMPL    R1, 4(SP)
                       0B     13   01A4   391         BEQL    5$
                  54   71     9A   01A6   392         MOVZBL  -(R1), R4
                  00'A544     95   01A9   393         TSTB    B^RES$PTAB(R5)[R4]              ; Might this be start of a double char?
                       F1     13   01AD   394         BEQL    2$
                       51     D6   01AF   395         INCL    R1
                  59   51     C2   01B1   396 5$:     SUBL2   R1, R9
                       09     13   01B4   397         BEQL    8$                              ; ONLY an optimization
                  50   59     C0   01B6   398         ADDL2   R9, R0
                  52   59     C0   01B9   399         ADDL2   R9, R2
                  53   59     C2   01BC   400         SUBL2   R9, R3
                                   01BF   401         ;
                                   01BF   402         ; Fetch the characters from strings
                                   01BF   403         ;
                                   01BF   404 8$:     FETCHSTR1
                                   01CD   405         FETCHSTR2
                                   01DB   406         ;
                                   01DB   407         ; They may differ.
                                   01DB   408         ; Are they different cases of the same char?
                                   01DB   409         ;
             10 6E     01     E0   01DB   410         BBS     #PSW$V_V, (SP), 16$
     0000'C549   0000'C544    91   01DF   411         CMPB    W^RES$UPPER(R5)[R4], W^RES$UPPER(R5)[R9]
                       05     13   01E8   412         BEQL    16$                             ; Yes, the binary compare is okay
```

```
                    6E    DC  01EA              MOVPSL  (SP)                            ; No, use this compare
              6E    02    C8  01EC              BISL2   #PSW$M_V, (SP)                  ; Don't do this again
                              01EF      16$:    ;
                              01EF              ; Fetch the collating values of the characters.
                              01EF              ;
          54     00'A544  9A  01EF              MOVZBL  B^RES$PTAB(R5)[R4], R4
                      2A  13  01F4              BEQL    101$
          59     00'A549  9A  01F6              MOVZBL  B^RES$PTAB(R5)[R9], R9
                      20  13  01FB              BEQL    202$
              59      54  91  01FD              CMPB    R4, R9
                      14  12  0200              BNEQ    150$
           04 AE      51  D0  0202      100$:   MOVL    R1, 4(SP)                       ; Establish a new stable point
                              0206              ;
                              0206              ; If characters remain in both strings, bypass the CMPC instruction
                              0206              ; if the first characters differ.  This is ONLY an optimization.
                              0206              ;
                              0206              ; This optimization was removed, to save code space.
                              0206              ;
                              0206              ;NOP
                              0206              ;
                              0206              ; Compare the remaining portions of the strings
                              0206              ;
  63    52  00'A5   61    50  2D  0206              CMPC5   R0, (R1), B^RES$PAD(R5), R2, (R3)
                      8E  12  020D              BNEQ    0$                              ; If equal, then R0 = 0
                 0203 8F  BA  020F      89$:    POPR    #^M<R0,R1,R9>                   ; Pop PSL & stable point, restore R9
                    FE50  31  0213              BRW     USETB1
                 0203 8F  BA  0216      150$:   POPR    #^M<R0,R1,R9>                   ; Pop PSL & stable point, restore R9
                    FE58  31  021A              BRW     NE
                    0093  31  021D      202$:   BRW     201$
          59     00'A549  9A  0220      101$:   MOVZBL  B^RES$PTAB(R5)[R9], R9
                              0225              ;
                              0225              ; R4 is zero; look for a double character.
                              0225              ; R9 contains the value from PTAB.
                              0225              ;
                      21  11  0225              BRB     112$
        59    59    F8 8F  78  0227      402$:   ASHL    #-8, R9, R9
                      D4  13  022C              BEQL    100$
                              022E              ;
                              022E              ; Fetch a character, and get it's collating value.
                              022E              ;
                              022E      111$:   FETCHSTR1NP     113$
                              0248      112$:   SPECDBL1
                      AC  13  0280              BEQL    111$
                      59  95  0282      113$:   TSTB    R9
                      2D  13  0284              BEQL    201$
              59      54  91  0286      301$:   CMPB    R4, R9
                      8B  12  0289              BNEQ    150$
        54    54    F8 8F  78  028B              ASHL    #-8, R4, R4
                      95  13  0290              BEQL    402$
        59    59    F8 8F  78  0292              ASHL    #-8, R9, R9
                      ED  12  0297              BNEQ    301$
                              0299      211$:   FETCHSTR2NP     301$
                              02B3      201$:   SPECDBL2
                      AC  13  02EB              BEQL    211$
                      97  11  02ED              BRB     301$
                              02EF              .DSABL  LSB
                              02EF
```

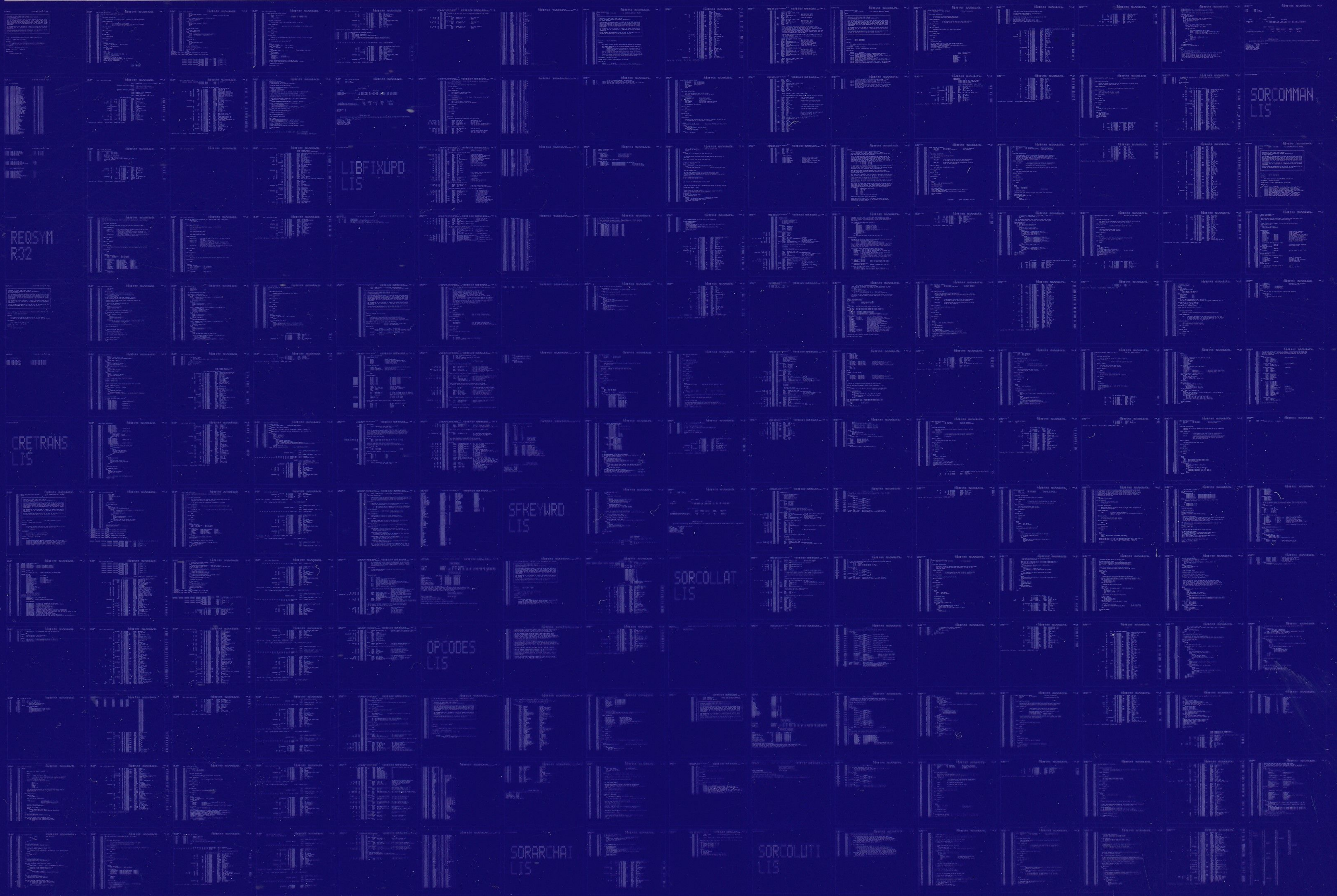
```
                02EF   470          .END
```

```
EQ                               0000007B R       02
GT                               00000077 R       02
JOIN_0                           00000018 R       02
JOIN_1                           00000096 R       02
LT                               0000006D R       02
NE                               00000075 R       02
POPNE                            00000071 R       02
PSW$M_V                        = 00000002
PSW$V_V                        = 00000001
PSW$V_Z                        = 00000002
RES$PAD                          ********      X  02
RES$PTAB                         ********      X  02
RES$REVERSE                      ********      X  02
RES$STAB                         ********      X  02
RES$TB                           ********      X  02
RES$UPPER                        ********      X  02
SOR$$COLLATE_0                   0000000A RG      02
SOR$$COLLATE_0_A                 00000000 RG      02
SOR$$COLLATE_1                   00000088 RG      02
SOR$$COLLATE_1_A                 0000007E RG      02
SOR$$COLLATE_2                   00000186 RG      02
USETB                            00000062 R       02
USETB1                           00000066 R       02

                                               +----------------+
                                               ! Psect synopsis !
                                               +----------------+

PSECT name                      Allocation          PSECT No.  Attributes
----------                      ----------          ---------  ----------
.  ABS  .                       00000000  (    0.)  00 (  0.)  NOPIC   USR   CON   ABS   LCL NOSHR NOEXE NORD   NOWRT NOVEC BYTE
$ABS$                           00000000  (    0.)  01 (  1.)  NOPIC   USR   CON   ABS   LCL NOSHR   EXE   RD     WRT NOVEC BYTE
SOR$RO_CODE                     000002EF  (  751.)  02 (  2.)    PIC   USR   CON   REL   LCL   SHR   EXE   RD   NOWRT NOVEC LONG

                                            +------------------------+
                                            ! Performance indicators !
                                            +------------------------+

Phase                      Page faults    CPU Time       Elapsed Time
-----                      -----------    --------       ------------
Initialization                      32    00:00:00.04    00:00:01.69
Command processing                 132    00:00:00.46    00:00:04.51
Pass 1                             138    00:00:02.15    00:00:08.60
Symbol table sort                    0    00:00:00.09    00:00:00.09
Pass 2                              92    00:00:00.93    00:00:03.99
Symbol table output                  4    00:00:00.03    00:00:00.03
Psect synopsis output                2    00:00:00.02    00:00:00.02
Cross-reference output               0    00:00:00.00    00:00:00.00
Assembler run totals               402    00:00:03.72    00:00:18.93

The working set limit was 1200 pages.
11496 bytes (23 pages) of virtual memory were used to buffer the intermediate code.
There were 10 pages of symbol table space allocated to hold 37 non-local and 62 local symbols.
470 source lines were read in Pass 1, producing 12 object records in Pass 2.
17 pages of virtual memory were used to define 15 macros.
```

```
                                             +------------------------------+
                                             ! Macro library statistics !
                                             +------------------------------+

Macro library name                           Macros defined
------------------                           --------------
_$255$DUA28:[SYSLIB]STARLET.MLB;2                      4

76 GETS were required to define 4 macros.

There were no errors, warnings or information messages.

MACRO/DISABLE=TRACE/LIS=LIS$:SORCOLLAT/OBJ=OBJ$:SORCOLLAT MSRC$:SORCOLLAT/UPDATE=(ENH$:SORCOLLAT)
```

0363 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY


SORCOMMAN
LIS

LIBFIXUPD
LIS

REQSYM
R32

CRETRANS
LIS

SFKEYWRD
LIS

SORCOLLAT
LIS

OPCODES
LIS

SORARCHAI
LIS

SORCOLUTI
LIS